

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

Controlled-Content Recoverable Blinded Certificates

Inventor(s):
Daniel R. Simon

ATTORNEY'S DOCKET NO. MS1-406US

005040 9903450
0543056 040500

005040 904560

1 **TECHNICAL FIELD**

2 This invention relates to cryptography. More particularly, the invention
3 relates to generating and using controlled-content recoverable blinded certificates.
4

5 **BACKGROUND OF THE INVENTION**

6 The number of people using computers, as well as the tasks they are used to
7 perform, is continually increasing. The Internet is one example of such an
8 increase – more and more people are communicating with one another,
9 researching information, and purchasing goods and services over the Internet.
10 However, accompanying this increasing usage of computers and the Internet is an
11 increasing concern about user-privacy, including concerns that individual's
12 purchasing and researching (or "web surfing") behavior is being monitored by
13 others.

14 A user can connect to the Internet at any time of day or night and purchase
15 electronic content that is immediately transferred to his or her computer (a process
16 referred to as "downloading"). Examples of such content include music (e.g.,
17 MP3 compressed audio files), text (e.g., electronic books), software applications,
18 etc. In order to obtain electronic content over the Internet, the seller or other
19 provider of such content often desires some assurances regarding the security of
20 the device requesting the content (e.g., the user's computer). Such assurances
21 indicate to the seller/provider that the electronic content obtained will not be used
22 inappropriately. For example, an assurance that music files transferred to the
23 device will not be improperly copied to another device.

24 Most users are willing to abide by such "proper usage" requirements for the
25 content they download. However, many are unwilling to forgo any personal

1 privacy in order to do so. For example, many users would be willing to accept a
2 mechanism that gave the seller/provider the desired assurances regarding the
3 security of their computer (or other device), but do not want their identity revealed
4 in order to do so.

5 One way in which these assurances can be made to the seller/provider is for
6 the requesting device to authenticate itself to the seller/provider. Such
7 authentication typically involves the requesting device identifying itself to the
8 seller/provider, either directly or indirectly via the authentication mechanism.
9 This identification, however, can also allow the user's behavior to be tracked. For
10 example, if a user continually uses the same public key for public key encryption
11 when obtaining electronic content, then the user can be tracked using that key.
12 Given the ability to track users using such mechanisms, they are unlikely to
13 achieve widespread user acceptance.

14 The invention described below addresses these disadvantages by providing
15 controlled-content recoverable blinded certificates.

16 17 **SUMMARY OF THE INVENTION**

18 In a cryptographic system, a certificate is used to provide information
19 regarding a client device. The certificate is blindly signed by a certifying authority
20 to preserve the anonymity of the client device. However, information is encoded
21 into the signature so that a content server can readily verify attributes of the client
22 device.

23 According to one aspect of the invention, a public key cryptographic
24 system is used in which the client device can generate new public/private key pairs
25 at will. A new public key is incorporated into a certificate and blindly signed by

000040 303460

1 the certifying authority. As the certificate is blinded, the certifying authority does
2 not know the exact content of what is being signed, but does encode into the
3 signature the same information as was in the signature of the last certificate it
4 signed for the client device. By changing public keys the client device can prevent
5 other devices from tracking it based on its public key, and by having the new
6 certificate (with the new public key) blindly signed the client device can prevent
7 the certifying authority from equating the previous public key to the new public
8 key.

9 According to another aspect of the invention, the client device is
10 recoverable from a failure of the client device during the process of obtaining a
11 new public key. Such a failure could result in a loss of the information used to
12 generate the new public keys. The client device can recover from such a failure by
13 using a fixed "pseudo-random" number generator to generate "random" numbers
14 based on a fixed seed value. The client device can readily re-generate its previous
15 public/private key pairs using this number generator. Each generated public key is
16 submitted to a certificate archive to determine whether a current valid certificate
17 exists for that public key. The generation and checking process continues until a
18 public/private key pair is generated for which a valid certificate exists.

19 20 **BRIEF DESCRIPTION OF THE DRAWINGS**

21 The present invention is illustrated by way of example and not limitation in
22 the figures of the accompanying drawings. The same numbers are used
23 throughout the figures to reference like components and/or features.

24 Fig. 1 shows a client/server network system and environment in accordance
25 with one embodiment of the invention.

1 Fig. 2 shows a general example of a computer that can be used in
2 accordance with the invention.

3 Fig. 3 illustrates an exemplary certificate that can be used in accordance
4 with the invention.

5 Fig. 4 is a block diagram illustrating an exemplary data flow for blindly
6 signing certificates and using the certificates to obtain electronic content according
7 to one embodiment of the invention.

8 Fig. 5 illustrates an example of the bit encoding according to one
9 embodiment of the invention.

10 Fig. 6 is a flowchart illustrating an exemplary process for generating a new
11 blindly signed certificate in accordance with one implementation of the invention.

12 Fig. 7 is a flowchart illustrating an exemplary process for obtaining content
13 using a new signed certificate in accordance with one implementation of the
14 invention.

15 Fig. 8 is a flowchart illustrating an exemplary process for restoring a key
16 pair generator to its proper state according to one embodiment of the invention.

17 18 19 **DETAILED DESCRIPTION**

20 The discussion herein assumes that the reader is familiar with cryptography.
21 For a basic introduction of cryptography, the reader is directed to a text written by
22 Bruce Schneier and entitled "Applied Cryptography: Protocols, Algorithms, and
23 Source Code in C," published by John Wiley & Sons with copyright 1994 (or
24 second edition with copyright 1996).
25

Network Structure

Fig. 1 shows a client/server network system and environment in accordance with one embodiment of the invention. Generally, the system includes one or more client devices 102, one or more content servers 104, one or more revocation servers 106, and one or more certifying authorities 108. The client devices 102, servers 104 and 106, and certifying authorities 108 communicate with one another over a data communications network. The data communications network in Fig. 1 is a public network 110, such as the Internet. The data communications network might also include local-area networks and/or private wide-area networks, and can include both wired and wireless sections. Alternatively, one or more of client devices 102, servers 104 and 106, and certifying authorities 108 may communicate with each other directly rather than via network 110.

Client device 102 includes an original certificate 112 that identifies various security-related attributes of client device 102. In the illustrated example, certificate 112 is part of a central processing unit (CPU) of client device 102 and is incorporated into the CPU when the CPU is manufactured (or alternatively when client device 102 is manufactured). Client device 102 also includes a current certificate 114. Current certificate 114 is initially the same as original certificate 112, but may be subsequently changed, as discussed in more detail below.

Client device 102 further includes a key pair generator 116 that generates a key pair 118 including a public key and a private key for the device 102. In the illustrated example, generator 116 generates a key pair in a conventional manner according to the well-known RSA (Rivest, Shamir, and Adelman) encryption technique. A random number seed 120 provides a fixed seed value to be used by

1 generator 116 in generating a key pair and recovering its state if necessary, as
2 discussed in more detail below.

3 Client device 102 also includes a public key 122 for certifying authority
4 108. Public key 122 allows client device 102 to establish a secure
5 communications link to certifying authority 108 via network 110, as discussed in
6 more detail below.

7 Content server 104 includes various electronic content 124 that can be
8 transferred to client 102 via network 110 (also referred to as "downloading").
9 Content 124 represents any type of electronic content, such as audio content (e.g.,
10 songs), video content (e.g., movies), textual content (e.g., articles, books,
11 magazines or newspapers), software (e.g., complete applications, upgrades, or
12 fixes), etc. Content 124 can include different titles (e.g., multiple different songs
13 and software applications) as well as different versions of the same content (e.g.,
14 different quality versions of the same song). To download content 124, client
15 device 102 requests the appropriate content 124 from content server 104. Based
16 on the requested content and the various attributes of client device 102 (e.g., its
17 security attributes), content server 104 determines which content (or which version
18 of particular content), if any, it will transfer to client device 102.

19 Certifying authority 108 certifies public keys generated by client device
20 102. This certification provides a verification to content server 104 that the public
21 key presented to server 104 by client 102 is actually from client 102 and that client
22 102 has the attributes that it claims to have. Certifying authority 108 includes a
23 secure connection module 126 to establish a secure connection to client 102 via
24 network 110, and a signature module 128 that certifies the public keys generated
25 by client device 102 by "signing" the keys, as discussed in more detail below.

1 Certifying authority 108 also includes a certificate archive 130 that is a record of
2 currently valid (non-revoked) certificates that have been signed by certifying
3 authority 108. A new certificate is added to certificate archive 130 and the
4 corresponding previous certificate removed from certificate archive 130 by client
5 device 102 (or alternatively certifying authority 108) when the new certificate is
6 created or signed. Certificate archive 130 can be used by client 102 in the event it
7 is recovering the state of key pair generator 116 or the current key pair 118, as
8 discussed in more detail below.

9 Revocation server 106 maintains a certificate revocation list 132 that
10 identifies revoked certificates. Certificates of client device 102, as well as other
11 client devices coupled to network 110, identify the public key and other attributes
12 of those devices. Certifying authority 108, in conjunction with client device 102,
13 can generate and certify a new certificate having a new public key. During the
14 certification process, client device 102 (or alternatively certifying authority 108)
15 also revokes the previous certificate used by client device 102. The previous
16 certificate is revoked so that the previous certificate (and thus the previous public
17 key) of client device 102 is no longer valid. Certificate revocation list 132 is a
18 record of these revoked certificates. Alternatively, previous certificates may not
19 be revoked.

21 Exemplary Computer Environment

22 In the discussion below, the invention will be described in the general
23 context of computer-executable instructions, such as program modules, being
24 executed by one or more conventional personal computers. Generally, program
25 modules include routines, programs, objects, components, data structures, etc. that

005040 9902450

1 perform particular tasks or implement particular abstract data types. Moreover,
2 those skilled in the art will appreciate that the invention may be practiced with
3 other computer system configurations, including hand-held devices,
4 multiprocessor systems, microprocessor-based or programmable consumer
5 electronics, network PCs, minicomputers, mainframe computers, and the like. In a
6 distributed computer environment, program modules may be located in both local
7 and remote memory storage devices.

8 Alternatively, the invention can be implemented in hardware or a
9 combination of hardware, software, and/or firmware. For example, the invention
10 can be implemented using one or more application specific integrated circuits
11 (ASICs).

12 Fig. 2 shows a general example of a computer 142 that can be used in
13 accordance with the invention. Computer 142 is shown as an example of a
14 computer that can perform the functions of client device 102, content server 104,
15 revocation server 106, or certifying authority 108 of Fig. 1. Computer 142
16 includes one or more processors or processing units 144, a system memory 146,
17 and a system bus 148 that couples various system components including the
18 system memory 146 to processors 144.

19 The bus 148 represents one or more of any of several types of bus
20 structures, including a memory bus or memory controller, a peripheral bus, an
21 accelerated graphics port, and a processor or local bus using any of a variety of
22 bus architectures. The system memory includes read only memory (ROM) 150
23 and random access memory (RAM) 152. A basic input/output system (BIOS) 154,
24 containing the basic routines that help to transfer information between elements
25 within computer 142, such as during start-up, is stored in ROM 150. Computer

1 142 further includes a hard disk drive 156 for reading from and writing to a hard
2 disk, not shown, a magnetic disk drive 158 for reading from and writing to a
3 removable magnetic disk 160, and an optical disk drive 162 for reading from or
4 writing to a removable optical disk 164 such as a CD ROM or other optical media.
5 The hard disk drive 156, magnetic disk drive 158, and optical disk drive 162 are
6 connected to the system bus 148 by an SCSI interface 166 or some other
7 appropriate interface. The drives and their associated computer-readable media
8 provide nonvolatile storage of computer readable instructions, data structures,
9 program modules and other data for computer 142. Although the exemplary
10 environment described herein employs a hard disk, a removable magnetic disk 160
11 and a removable optical disk 164, it should be appreciated by those skilled in the
12 art that other types of computer readable media which can store data that is
13 accessible by a computer, such as magnetic cassettes, flash memory cards, digital
14 video disks, random access memories (RAMs) read only memories (ROM), and
15 the like, may also be used in the exemplary operating environment.

16 A number of program modules may be stored on the hard disk, magnetic
17 disk 160, optical disk 164, ROM 150, or RAM 152, including an operating system
18 170, one or more application programs 172, other program modules 174, and
19 program data 176. A user may enter commands and information into computer
20 142 through input devices such as keyboard 178 and pointing device 180. Other
21 input devices (not shown) may include a microphone, joystick, game pad, satellite
22 dish, scanner, or the like. These and other input devices are connected to the
23 processing unit 144 through an interface 182 that is coupled to the system bus. A
24 monitor 184 or other type of display device is also connected to the system bus
25 148 via an interface, such as a video adapter 186. In addition to the monitor,

1 Generally, the data processors of computer 142 are programmed by means
2 of instructions stored at different times in the various computer-readable storage
3 media of the computer. Programs and operating systems are typically distributed,
4 for example, on floppy disks or CD-ROMs. From there, they are installed or
5 loaded into the secondary memory of a computer. At execution, they are loaded at
6 least partially into the computer's primary electronic memory. The invention
7 described herein includes these and other various types of computer-readable
8 storage media when such media contain instructions or programs for implementing
9 the steps described below in conjunction with a microprocessor or other data
10 processor. The invention also includes the computer itself when programmed
11 according to the methods and techniques described below. Furthermore, certain
12 sub-components of the computer may be programmed to perform the functions
13 and steps described below. The invention includes such sub-components when
14 they are programmed as described. In addition, the invention described herein
15 includes data structures, described below, as embodied on various types of
16 memory media.

17 For purposes of illustration, programs and other executable program
18 components such as the operating system are illustrated herein as discrete blocks,
19 although it is recognized that such programs and components reside at various
20 times in different storage components of the computer, and are executed by the
21 data processor(s) of the computer.

22 23 **Client Certificates**

24 Client device 102 can provide information about itself to content server 104
25 via a certificate previously signed by a certifying authority. Fig. 3 illustrates an

exemplary certificate 210 that includes a public key 212 and one or more attributes 214. Public key 212 is the current public key (in accordance with RSA cryptography) being used by client device 102. Attributes 214 identify various characteristics of client device 102, such as what type of device client 102 is or how client 102 was constructed. For example, attributes may identify client device 102 as a particular type of device (e.g., a personal computer with an Intel® microprocessor, a personal MP3 audio player, or an Internet browsing device such as a WebTV® terminal or a gaming console), or a device with particular security characteristics (e.g., a computer with an Intel® microprocessor built to run only the Microsoft® Windows NT® operating system). In the illustrated example, attributes 214 include a security level 216 of client device 102 (e.g., a numeric level of a predefined set of security levels) and an identifier 218 of certifying authority 108. Additionally, an expiration date and time 220 may be included in attributes 214, identifying when certificate 210 will expire (no longer be valid).

Client device 102 includes a processor (such as processing unit 144 of Fig. 2) that is capable of performing cryptographic functions, such as signing, encrypting, decrypting, and authenticating. An additional cryptographic accelerator (not shown) may also be included to assist the processor with intensive mathematical computations commonly involved in cryptographic functions.

The processor manufacturer equips processor 144 with a pair of public and private keys 118 that are unique to the processor 144, and thus unique to device 102. Other physical implementations may include storing the key on an external device to which the main processor has privileged access (where the stored secrets are inaccessible to arbitrary application or operating system code). The private

1 key is never revealed and is used only for the specific purpose of signing stylized
2 messages, as discussed below in more detail.

3 The processor manufacturer also issues a signed original certificate 112
4 testifying that it produced the processor according to a known specification and
5 including the attributes 214 of Fig. 3. Generally, the certificate 112 testifies that
6 the manufacturer created the key pair 118, placed the key pair onto the processor
7 144, and then destroyed its own knowledge of the private key, or alternatively that
8 the private key was generated internally in the device, and the public key was
9 obtained from it under controlled circumstances by the manufacturer. In this way,
10 nobody but the client device 102 knows the client device private key; the same key
11 is not issued to other processors. The certificate can in principle be stored on a
12 separate physical device but still logically belongs to the processor with the
13 corresponding key. Alternatively, the manufacturer of client device 102 rather
14 than the processor 144 may equip client device 102 with the public/private key
15 pair 118 and certificate 112.

16 In order for the client device 102 to cryptographically sign a message (e.g.,
17 a certificate), its public key (from key pair 118) is made known to the device(s)
18 that will receive the message. The public key can be included in non-encrypted
19 form along with the signed message, or may otherwise be made publicly known.
20 Making the public key publicly known, however, can result in the loss of at least
21 some anonymity of the client device 102. If the client device 102 continually uses
22 the same public key, then that public key can become associated with device 102
23 and the transactions conducted, information obtained, etc. can be tracked based on
24 the public key.
25

005040" 3030450

1 The invention solves this problem by allowing client device 102 to change
2 key pair 118. A new key pair can be generated by generator 116 and the new
3 public key incorporated into a certificate that is forwarded to certifying authority
4 108 for certification. The certificate is blinded so that certifying authority 108
5 does not know the value of the new public key and thus cannot associate the new
6 public key with the previous public key. However, the certifying authority 108
7 digitally signs the certificate from client device 102 and encodes some (or all) of
8 the same attributes into the new certificate (with the new public key) as were
9 associated with the previous certificate used by client device 102. Thus, client
10 device 102 is able to generate a new key pair and have the public key certified as
11 having the same attributes as the previous key without revealing any information
12 regarding the identity of client device 102.

13 Fig. 4 is a block diagram illustrating an exemplary data flow for blindly
14 signing certificates and using the certificates to obtain electronic content according
15 to one embodiment of the invention. The client device 102 and certifying
16 authority 108 establish a secure connection 232 between themselves (e.g., via
17 network 110 of Fig. 1). The client device 102 generates a new key pair,
18 incorporates the new public key into a new certificate, blinds the new certificate,
19 and transmits the blinded certificate 234 to certifying authority 108 via the secure
20 connection 232. Client device 102 also requests that certifying authority 108 sign
21 the certificate indicating that client device 102 has all or many of the same
22 attributes as the previous certificate used by client device 102. Certifying
23 authority 108 verifies that the blinded certificate 234 is to have the same attributes
24 as the previous certificate based on information encoded in the previous signed
25 certificate. If certifying authority 108 can verify such, then it issues a new signed

certificate 236 for blinded certificate 234; otherwise it will not issue signed certificate 236.

Signed certificate 236 is received by client 102 and stored as current certificate 114 of Fig. 1. Signed certificate 236 gives client 102 a valid certificate in which is encoded various attributes. However, since certifying authority 108 issued signed certificate 236 based on a blinded certificate, certifying authority 108 has no knowledge of what the public key in that certificate is.

Client 102 can then use its new public key to obtain electronic content from content server 104. Client 102 forwards the current certificate and a request for content 238 to content server 104. Alternatively, a secure connection between client 102 and content server 104 may be established analogous to secure⁹ connection 232 between client 102 and certifying authority 108. Content server 104 evaluates the request and certificate 238 to determine what content, if any, to deliver to client 102 and/or how to deliver the content to client 102 (e.g., what fee to charge, how to collect the fee, what additional security precautions to insist on, etc.). This determination is made by content server 104, at least in part, by evaluating the attributes encoded in the certificate received from client 102. Based on its evaluation, content server 104 forwards the appropriate requested content 240 in the appropriate manner to client device 102.

The invention makes use of public key cryptography to encrypt and decrypt information as well as to digitally sign and verify messages. The invention is described using the well-known RSA algorithm. Alternatively, other public key cryptographic algorithms could be used, such as well-known elliptic curve cryptosystems or well-known Diffie-Hellman key agreement protocols.

Secure connection 232 is established by client 102 and certifying authority 108 generating a session key. According to one implementation, the session key is generated using the client public key and the certifying authority public key. The client 102 selects a random value, encrypts the random value with the certifying authority public key 122 of Fig. 1, and sends the encrypted random value to certifying authority 108 along with current certificate 114 (which includes the client public key). The certifying authority 108 also selects a random value, encrypts the random value with the client public key, and sends the encrypted random value to client 102. Each of the client 102 and certifying authority 108 can decrypt the encrypted random values they receive using their respective private keys. The client 102 and certifying authority 108 then combine the two random values in some known manner (e.g., adding the two values, concatenating the two values, etc.) to generate the session key. All subsequent communications between client 102 and certifying authority 108 via secure connection 232 are encrypted in a conventional manner using this session key.

The public and private keys are generated and used in accordance with RSA. Using RSA, two large prime numbers p and q are selected and multiplied to generate a product n . A value e is also generated and is relatively prime to $(p-1)(q-1)$. A value of d , which is the inverse of e is also determined, such that:

$$ed = 1 \bmod (p-1)(q-1)$$

The private key then is the pair p and q , or alternatively d , and the public key is the pair n and e .

A message m (e.g., a series of numbers representing a textual message, such as the ASCII values for alphanumerics) can be encrypted to generate m_{encrypt} via the following formula:

$$m_{encrypt} = m^e \pmod n$$

The intended recipient of the message knows the private key and can easily decrypt the message using the following formula:

$$m = (m_{encrypt}^e)^d \pmod n$$

However, no known algorithm can efficiently decrypt the encrypted message $m_{encrypt}$ without knowing the private key.

A message m can similarly be digitally signed by the owner of the private key to generate m_{signed} via the following formula:

$$m_{signed} = m^d \pmod n$$

Anyone else that knows the public key can decrypt the signed message m_{signed} and verify that it was indeed signed using the private key corresponding to the public key using the following formula:

$$m = (m_{signed}^d)^e \pmod n$$

The invention can also have a message m be blindly signed. A blindly signed message is one that is digitally signed without the device doing the signing having any knowledge of the underlying message. Thus, client 102 can have certifying authority 108 sign a message m (e.g., including a new public key) without certifying authority 108 having any knowledge of the actual content of message m (e.g., the new public key). The blinding process is carried out by client 102 generating a value x and multiplying the message m by the value x^e . The blinded message mx^e is then sent to certifying authority 108, which signs the blinded message according to the following formula:

$$mx_{signed} = (mx^e)^d \pmod n$$

This value is returned to client 102, which can easily generate the signed message $(m^d \pmod n)$ according to the following:

1 integers which were not included in e . Each message (e.g., a certificate) would
 2 then be signed twice, once with the value of e and once with the value of e_2 . Both
 3 of these signed messages would then be verified by content server 104 in
 4 determining whether to provide content to client device 102.

5 The values of the integers e^i should be chosen to be relatively prime to $(p-$
 6 $1)(q-1)$. This can be accomplished, for example, by choosing values of p and q
 7 such that $(p-1)$ and $(q-1)$ are both twice a prime number, and skipping the value
 8 two when choosing e^i values.

9 Additionally, with the value of e selected, the corresponding value of d ,
 10 which is a product of multiple integers d^i , can be readily determined. The value of
 11 each integer d^i can be determined by calculating the value such that $e^i d^i = 1 \bmod (p-$
 12 $1)(q-1)$.

13 Fig. 5 illustrates an example of the bit encoding according to one
 14 embodiment of the invention. In the illustrated example, up to fifteen different
 15 integers e^i are included. In the example encoding 248, the first, fourth, ninth and
 16 twelfth integers e^i are included, resulting in the encoded value of
 17 100100001001000. In the example encoding 250, the second, third, fifth, sixth,
 18 seventh, eighth, tenth, eleventh, thirteenth, fourteenth, and fifteenth integers e^i are
 19 included, resulting in the encoded value of 011011110110111.

20 The information encoded into the digital signature identifies various
 21 attributes of client device 102. In one implementation, each of the attributes 216,
 22 218, and 220 included in certificate 210 of Fig. 3 are encoded into the digital
 23 signature. The encoding can further be seen from the following example. Assume
 24 that eight different security levels are predefined, one of which is identified as
 25 security level 216 of certificate 210. Three different values e^i (e.g., e^1 , e^2 , and e^3)

are used to encode the security level 216 into the signature. Thus, assuming that a security level of five would be encoded as its binary representation (101₂), then the values encoded into the digital signature as e^1 , e^2 , and e^3 would be 1, 0, and 1, respectively.

By way of further example, assume that the values of the fifteen integers e^i are as indicated in Table I below.

Table I

Integer	Value
e^1	3
e^2	5
e^3	7
e^4	11
e^5	13
e^6	17
e^7	19
e^8	23
e^9	29
e^{10}	31
e^{11}	37
e^{12}	41
e^{13}	43
e^{14}	47
e^{15}	53

Using the values listed in Table I, the value of encoding 248 would be $3 \cdot 11 \cdot 29 \cdot 41$, the product of which is 39,237. Similarly, the value of encoding 250 would be $5 \cdot 7 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 31 \cdot 37 \cdot 43 \cdot 47 \cdot 53$, the product of which is 11,223,947,638,085.

Certifying authority 108 uses this value e with encoded information to sign the blind certificate it receives from client device 102. Certifying authority 108

000040-950456

1 generates the signature for the new certificate based on the encoded information in
2 the previous certificate's signature (the previously signed certificate from client
3 device 102 was received during the establishment of secure connection 232 of
4 Fig. 4). For the first certificate signed by certifying authority 108 for client device
5 102, there is no previously signed certificate. Thus, certifying authority 108 relies
6 on the digital signature of the manufacturer on original certificate 112 of Fig. 1 to
7 verify the certificate and encodes the attributes from original certificate 112 into
8 the signature. Certifying authority 108 can thus ensure that it does not attribute
9 any greater level of security to the device when encoding information into the new
10 certificate than was encoded into the previous certificate.

11 Certifying authority 108 may, however, modify the information encoded in
12 the new signature. For example, an expiration date and time for the certificate
13 may be encoded into the signature, indicating that the certificate is valid for a
14 period of six months after issuance. Thus, each time the client device 102 requests
15 a new signed certificate, certifying authority 108 encodes the new expiration date
16 and time into the new signature.

17 In order to obtain content from content server 104, client device 102
18 forwards a request for content and its current signed certificate 238 to content
19 server 104. The signed certificate makes a representation to content server 104 of
20 various attributes of client device 102. Content server 104 verifies these attributes
21 using the information encoded into the digital signature.

22 Both the public key portion n and the encodings for e used by certifying
23 authority 108 are made known to content server 104. The public key portion n and
24 the encodings for e can be made publicly known, or alternatively can be
25 communicated securely (e.g., using public-key encryption) to content server 104

from certifying authority 108. Given the attributes that client device 102 claims to have, and the encodings used by certifying authority 108 in signing client certificates, content server 104 can readily determine whether the attributes encoded in the digital signature match the claimed attributes. For example, using the encodings a proposed value of e can be generated based on the claimed attributes. This proposed value of e can then be used, in conjunction with the public key portion n received from certifying authority 108, to verify the signature of the certificate.

If the claimed attributes are not verified, then content server 104 can refuse to provide content to client device 102. However, if the claimed attributes are verified, then content server 104 transmits the requested content 240 to client device 102. What content and/or how the content is transmitted to client device 102 can vary depending on the attributes of client device 102. For example, a fee may or may not be imposed, a particular quality level of content may be transmitted, confirmation from client device 102 of additional security precautions that must be taken, a licensing agreement accepted, etc.

Fig. 6 is a flowchart illustrating an exemplary process for generating a new blindly signed certificate in accordance with one implementation of the invention. Steps on the left side of Fig. 6 are implemented by client device 102 of Figs. 1 and 4, while steps on the right side of Fig. 6 are implemented by certifying authority 108 of Figs. 1 and 4. The process of Fig. 6 may be performed in software. Fig. 6 is described with additional reference to components in Figs. 1 and 4.

Initially, client device 102 and certifying authority 108 establish a secure connection between each other (steps 252 and 254) based on the client's previous certificate (so that the certifying authority knows the correct attributes to encode

1 into the new certificate). Client 102 then constructs a new public/private key pair
2 (step 256). Client 102 then constructs a new certificate by replacing the public
3 key in its current certificate with the new public key, and blinds the new certificate
4 (step 258). Client 102 sends the blinded certificate to certifying authority 108
5 (step 260). Certifying authority 108 receives the blinded certificate (step 262) and
6 signs the blinded certificate encoding attributes into the signature based on the
7 previous certificate (step 264).

8 Certifying authority 108 then sends the signed blinded certificate to client
9 device 102 (step 266). Client device 102 receives the signed blinded certificate
10 (step 268), and unblinds the signed blinded certificate to generate the signed
11 certificate (step 270).

12 Fig. 7 is a flowchart illustrating an exemplary process for obtaining content
13 using a new signed certificate in accordance with one implementation of the
14 invention. Steps on the left side of Fig. 7 are implemented by client device 102 of
15 Figs. 1 and 4, while steps on the right side of Fig. 7 are implemented by content
16 server 104 of Figs. 1 and 4. The process of Fig. 7 may be performed in software.
17 Fig. 7 is described with additional reference to components in Figs. 1 and 4.

18 Initially, client device 102 generates a content request (step 282). This may
19 be automatically generated by client device 102 or alternatively may be in
20 response to a user request at client device 102. Client 102 sends the request and
21 current certificate 114 (signed by certifying authority 108) to content server 104
22 (step 284).

23 Content server 104 receives the request and signed certificate (step 286),
24 and identifies the attributes encoded in the signature (step 288). Content server
25 104 then checks whether the attributes in the certificate match the attributes

005040 950450

To generate a new key pair, generator 116 generates two random integers (via the pseudo-random number generator) and begins testing each integer, as well as the successors of each, for primality. The order in which successors are tested is fixed (and thus can be subsequently duplicated if necessary, as discussed in more detail below). Any of a wide variety of conventional techniques can be used to test for primality, such as the well-known Miller-Rabin or Solovay-Strassen techniques. Once a prime number has been identified from each sequence (which are the values p and q of the private key), the value of n for the public key can be readily generated by multiplying the two prime numbers.

Situations can arise where client 102 needs to re-generate previously generated public/private key pairs. For example, a failure of client device 102 at an inopportune moment may cause client 102 to lose its private key (e.g., erased from memory) and/or the random number generator to lose its current state. In such situations, client 102 re-starts its key pair generation process with seed value 120 to bring key pair generator 116 back to its previous state.

Fig. 8 is a flowchart illustrating an exemplary process for restoring key pair generator 116 to its proper state according to one embodiment of the invention. The process of Fig. 8 is implemented by client device 102 of Figs. 1, and may be performed in software. Fig. 8 is described with additional reference to components in Fig. 1.

Initially, generator 116 generates a public/private key pair based on its seed value 120 (step 312). Because the seed value 120 has not changed and the algorithm(s) for generating the prime numbers of the private key are fixed, each time that generator 116 generates a public/private key pair using its seed value 120 the same public/private key pair will be generated. Generator 116 then requests a

005040 9506456

1 valid certificate for the generated public key (step 314). This request is made to a
2 certificate server or archive, which is archive 130 in the illustrated example of
3 Fig. 1.

4 The certificate server or archive will check whether a valid certificate exists
5 for the public key and return either the valid certificate or an indication that no
6 such certificate exists. Generator 116 checks whether it receives a valid certificate
7 or an indication that no such certificate exists (step 316). If a valid certificate
8 exists, then its state has been restored (step 318). However, if no such certificate
9 exists then generator 116 generates another public/private key pair (step 320).
10 Eventually, a valid certificate will be received and the state of generator 116
11 restored (step 318).

12 13 Conclusion

14 Thus, the invention provides controlled-content recoverable blinded
15 certificates. A client device can readily create a new public key with a digital
16 signature from a certifying authority and with security attributes of the client
17 device advantageously encoded into the digital signature. A blinding process is
18 used to blind the public key for signature, thereby preventing the certifying
19 authority from tracking the client device based on the relationship between its
20 previous public key and its new public key. A content server can use these
21 encoded security attributes to determine whether and how to provide requested
22 electronic content to the client device. Additionally, according to one aspect of
23 the invention, the client device uses a fixed algorithm and a fixed seed value to
24 generate public/private key pairs, thereby advantageously allowing the client
25 device to recover from a failure during the process of changing its public key.

005040" 33334560

1 Although the invention has been described in language specific to structural
2 features and/or methodological steps, it is to be understood that the invention
3 defined in the appended claims is not necessarily limited to the specific features or
4 steps described. Rather, the specific features and steps are disclosed as preferred
5 forms of implementing the claimed invention.